

SQL Structured Query Language

Data: is a raw fact which describes attributes of an Entity

Attributes - Properties
Entity - Object

Entity - is nothing but an object it may be either living thing or a non-living thing.

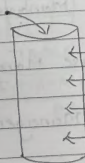
- Anything which has physical existence in a real world is known as Entity.

Database - is a place medium or a container used to store data in a systematic & organized manner.

- In Database we can perform several operations.
- The operations for universally called as CRUD operations.

4 tasks / operation

CRUD



Database

Page No. _____
Date _____

SQL Structured Query Language

Data: is a row fact which describes attributes of an entity

Attributes - Properties

Entity - Object

Entity - is nothing but an object it may either living thing or a non-living thing.

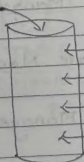
- Anything which has physical existence in a real world is known as an entity.

DataBase - is a place medium or a container used to store data in a systematic & organized manner.

- In Database we can perform several operations.
- The operations for universally known as CRUD operations.

4 tasks / operation

CRUD



Create / Insert

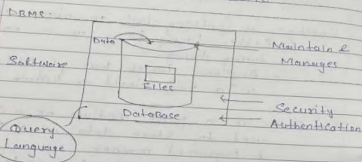
Read / Retrieve

Update / Modify

Drop / Delete

DataBase

- * DBMS (Database Management System) is a software which is used to maintain and manage the software database.
- DBMS has two important features: Security and Authorization.
 - In DBMS we can store data in the form of files.
 - We are using query language to communicate with DBMS.



Types of DBMS:-

- Hierarchical Database Management System [HDBMS]
- Network Database Management System [NDBMS]
- Object oriented Database Management System [OODBMS]
- Relational Database Management System [RDBMS]

- * Relational Model:- Was designed by Data Scientist E.F. Codd.
- Any DBMS which supports Relational model there will become RDBMS.

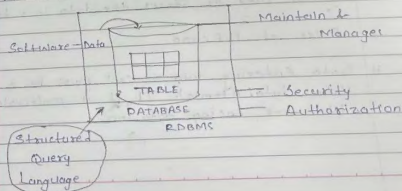
$$[DBMS] + [Relational Model] \rightarrow [RDBMS]$$

- Any DBMS which follows the rules of E.F. Codd is known as RDBMS.

$$[DBMS] + [Rules of E.F. Codd] \rightarrow [RDBMS]$$

- * RDBMS [Relational Database Management System]
- RDBMS is a type of DBMS software which is used to maintain & manage the database.
 - In RDBMS we can store data in the form of tables [Rows & Columns].
 - We are using Structured Query Language to communicate with RDBMS.

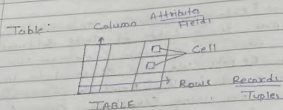
RDBMS:



* Difference b/w DBMS & RDBMS

DBMS	RDBMS
- Database Management System	- Relational DBMS
- We can store Data in the form of Files	- We can store Data in the form of Tables
- We are using Query lang. to Communicate with DBMS	- We are using Structured Query lang. to Communicate with RDBMS.

* Table:- is a logical organization of Rows & Columns and Cells.



* Cell:- is a intersection between rows & Columns.
- Cell is smallest block or a unit which is used to store the data in a table.

* Rules of E.F.Codd

- i) Data Entering into a Cell must be a Single Value data (Atomic). If it is multivalued we are facing data loss.

- ii) In RDBMS Everything can be stored in the form of table including MetaData.

Metadata:- Details about the Data is known as Metadata.

:- Metadata are stored in a table called MetaTable

:- This Metatable are autogenerated.

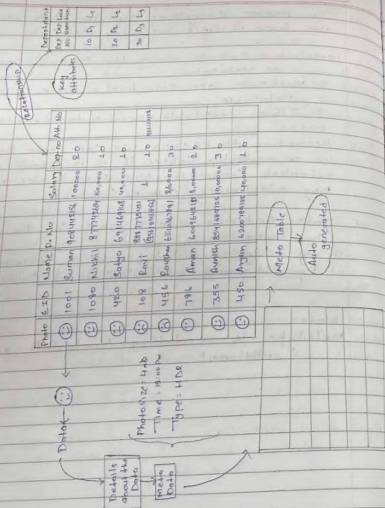
- iii) According to E.F.Codd we can store Data in a multiple tables. If it is needed establish Connection b/w the tables using key attributes (Common Column).

- iv) Data Entering into a table can be validated by two steps

- i) By assigning datatypes to a Columns
- ii) By assigning Constraints to a Columns.

Note:-

Datatype is mandatory Constraints are optional.



* Datatypes:- It is used to describe which kind or type of data should be entering into a particular column or a memory allocation.

* Types of Datatypes:-

- i) char
- ii) VARCHAR/VARCHAR2 and DATE
- iii) NUMBER
 - ↳ Character Large object
 - ↳ Binary Large object

i) char

- 'A-Z', 'a-z', '0-9', Special character, Alphanumeric (@, !, %, #, -)

- Characters are must Enclosed with the Single quotations (' ').

Syntax

Char(size)

The max. size of char data type is 2000 characters

- It follows fixed length of memory allocations

Eg: 'KRISHNA'

Char(10)

K R I S H N A

Used Memory

Unused Memory

Fixed length of memory Allocation

- In char datatype unused memory will be wastage of memory.

- Sql is not a Case Sensitive language But Characters data are Case Sensitive.

(iii) VARCHAR2

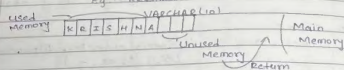
- 'a-z', '0-9', special char. 'Alphanumeric' (0, 1, \$, %, -)

- Characters are must enclosed with the single quotation ('').

- Syntax

The max size of varchar data type is 2000.

- It follows variable length of memory allocation.
eg: 'KRISHNA'



VARCHAR2 - It is updated version of VARCHAR.

Syntax

VARCHAR2(size)

In VARCHAR2 we can put too characters as a max size.

* Difference b/w Char & VARCHAR2

Char	VARCHAR2
Char (size)	VARCHAR (size) VARCHAR2 (size)
2000 ch max size	4000 ch max size
Fixed length	Variable length

(iv) DATE: It is used to store only the Dates.

Syntax: Date

- We can store Dates in a Oracle specified format.

Oracle format

- 'DD-MON-YY'

'01-JAN-25'

- 'DD-MON-YYYY'

'31-DEC-2025'

(v) NUMBER: It is used to store only the numeric Value.

Syntax: Number (Precision, [Scale]).

- In Number Datatype we can use 2 args
i) Precision

ii) Scale

- Precision is mandatory & Scale is optional.

Precision: It is used to allocate no. of digits required to store an int. values.

- The Range of precision is 1 to 38 Digits.

Scale: - It is used to store decimal Values within the given precision.

- The range of Scales is -84 to 127.

- The default Value of Scale is 0.

eg-

Number(6): 111111

Number(5,3): 111.111

Number(7,3): 1.111

Number(3,5): 1.00111

Number(8,10): 1.0011111111

1) Large objects:-

↳ Char Large object:- It is used to store huge amount of characters upto 4GB of size.
Syntax: CLOB

↳ Binary Large object:- It is used to store the Binary Values of photo, video, audio, documents etc upto 4GB of size.
Syntax: BLOB

* Constraints:- are the rules or a Condition which are assigned to the Column for Validation

Types of Constraints:-

- 1) Unique
- 2) Not Null
- 3) Check
- 4) Primary Key
- 5) Foreign Key

1) Unique:- It is a Constraints it Cannot Accept duplicate or repeated Values.

2) Not NULL:- It is a Constraints it Cannot Accept any Null Values.

Null:- It is nothing But Empty Cell.
- It Cannot occupies any memory space.
- 0 and space Cannot be a Null.
- If we perform any Arithmetic operation with Null it returns Null Value.
e.g. $100 + \text{Null} \rightarrow \text{Null}$

3) Check:- It is a Constraints which gives extra Validation with the help of Conditions. If the Conditions get satisfied the Value or data get Entered into a column or Else it rejects the data.
e.g. $\text{Check}(\text{Salary} > 0)$
 $5000 > 0$
- $\text{Check}(\text{Length}(\text{Phone_no}) > 10)$
 $2-9808420121$
 $10 > 10$

Note

Length(): It is used to Count no. of digits or Characters.

4) Primary Key:- It is used to identify the unique records from the table.

Characteristics of primary keys:-

- We have only one primary Key in a table.
- Primary Key Cannot accept duplicate or repeated Values.
- Primary Key Cannot accept null Values.
- Primary Key is a Combination of Unique & Not Null Constraints.

- Primary key is not mandatory But it is highly recommended to use the table

1) Foreign Keys - It is used to Establish Connection b/w the tables

Characteristics of foreign keys:-

- table have n numbers of foreign keys in a table.
- Foreign key will accept duplicate or repeated values.
- Foreign key can accept Null values.
- Foreign key is not a combination of Unique and Not Null Constraints.
- Foreign attributes to become a foreign key it must be primary key in its own table.
- foreign key presenting child table But it actually belongs to parent table.
- Foreign key is also known as referential integrity Constraints.

Primary Key	Foreign Key
- It Contains unique Value	- It Contains duplicate Value
- It Cannot Contain null Value	- It Contain null value
- The table has only 1 primary key	- There Can be n no. of foreign key
- used to identify the unique records from the table	- used to Establish Connection b/w the table.

Department

Dept. Description	Dept. No.
Accounting	10
Marketing	20
Production	30
Research & Development	40
Sales	50
Support	60
Training	70
Human Resources	80
Information Systems	90
Legal	100
Other	110

Employee

Employee Name	Employee ID
James	101
Scott	102
Deena	103
John	104
Paul	105
Christy	106
David	107
Michael	108
Wendy	109
Pat	110
Allen	111
Kim	112
Tim	113
Lisa	114
Tracy	115
Ruth	116
Cheryl	117
David	118
Neena	119
Irene	120
Timothy	121
Jonathan	122
Vanessa	123
Anthony	124
Jeffrey	125
Art	126
Shelley	127
Greg	128
Bryant	129
Michelle	130
Patricia	131
John	132
Yusef	133
John	134
John	135
John	136
John	137
John	138
John	139
John	140
John	141
John	142
John	143
John	144
John	145
John	146
John	147
John	148
John	149
John	150

Employee

Emp No.	Name	Salary	Dept	Grade	Emp No.	Name	Salary	Dept	Grade
101	James	9500	10	10	101	James	9500	10	10
102	Scott	7500	10	10	102	Scott	7500	10	10
103	Deena	6500	10	10	103	Deena	6500	10	10
104	John	5500	10	10	104	John	5500	10	10
105	Paul	4500	10	10	105	Paul	4500	10	10
106	Christy	3500	10	10	106	Christy	3500	10	10
107	David	2500	10	10	107	David	2500	10	10
108	Michael	1500	10	10	108	Michael	1500	10	10
109	Wendy	1200	10	10	109	Wendy	1200	10	10
110	Pat	900	10	10	110	Pat	900	10	10
111	Allen	800	10	10	111	Allen	800	10	10
112	Kim	700	10	10	112	Kim	700	10	10
113	Tim	600	10	10	113	Tim	600	10	10
114	Lisa	500	10	10	114	Lisa	500	10	10
115	Tracy	400	10	10	115	Tracy	400	10	10
116	Ruth	300	10	10	116	Ruth	300	10	10
117	Cheryl	200	10	10	117	Cheryl	200	10	10
118	David	100	10	10	118	David	100	10	10
119	Neena	900	10	10	119	Neena	900	10	10
120	Irene	800	10	10	120	Irene	800	10	10
121	Timothy	700	10	10	121	Timothy	700	10	10
122	Jonathan	600	10	10	122	Jonathan	600	10	10
123	Vanessa	500	10	10	123	Vanessa	500	10	10
124	Anthony	400	10	10	124	Anthony	400	10	10
125	Jeffrey	300	10	10	125	Jeffrey	300	10	10
126	Art	200	10	10	126	Art	200	10	10
127	Shelley	100	10	10	127	Shelley	100	10	10
128	Greg	900	10	10	128	Greg	900	10	10
129	Bryant	800	10	10	129	Bryant	800	10	10
130	Michelle	700	10	10	130	Michelle	700	10	10
131	Patricia	600	10	10	131	Patricia	600	10	10
132	John	500	10	10	132	John	500	10	10
133	Yusef	400	10	10	133	Yusef	400	10	10
134	John	300	10	10	134	John	300	10	10
135	John	200	10	10	135	John	200	10	10
136	John	100	10	10	136	John	100	10	10
137	John	900	10	10	137	John	900	10	10
138	John	800	10	10	138	John	800	10	10
139	John	700	10	10	139	John	700	10	10
140	John	600	10	10	140	John	600	10	10
141	John	500	10	10	141	John	500	10	10
142	John	400	10	10	142	John	400	10	10
143	John	300	10	10	143	John	300	10	10
144	John	200	10	10	144	John	200	10	10
145	John	100	10	10	145	John	100	10	10
146	John	900	10	10	146	John	900	10	10
147	John	800	10	10	147	John	800	10	10
148	John	700	10	10	148	John	700	10	10
149	John	600	10	10	149	John	600	10	10
150	John	500	10	10	150	John	500	10	10

* Difference b/w Primary key & foreign key

PRIMARY KEY	FOREIGN KEY
→ A primary key is used to ensure data in the specific column is unique	→ A foreign key is a column or a group of columns in a relational database table that provide link b/w data in 2 tables
→ It uniquely identifies a record in relational database table	→ It refers to the field in a table which is the primary key of another table
→ Only one primary key is allowed in a table	→ More than one foreign key is allowed in a table
→ It is a combination of UNIQUE and NotNull constraints	→ It can contain duplicate values and a table in a relational database
→ It does not allow NULL value	→ It can also contain NULL value
→ Its value cannot be deleted from the parent table	→ Its value can be deleted from the child table

* Difference b/w unique and primary

Unique	Primary
- Cannot accept Duplicate	- Cannot Accept Duplicate
- No Combination	- Combination of Unique & Null
- It can identify but they cannot be duplicate	- It can be identity only item
→ Does not Allow Null Value	- Does not Allow Null Value
→ Can have multiple unique constraint in a table	- Only 1 primary key per table
→ Ensure values in a column are unique	- used to identify the unique records from table

* Overview of sql statement

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Transaction Control Language (TCL)
- Data Control Language (DCL)
- Data Query Language (DQL)
- It is used to retrieve or fetch data from the DataBase

DDL statements:-

- Select - Display
- Projection - only Column
- Selection - Both Row & Column
- Joins - Data from multiple table

Select:- It is used to retrieve data from the table and display it.

Projection:- It is used to retrieve data from the table by selecting only the columns.

- In projection if the columns are get selected by default the values present in that column also get selected.

Syntax:-

SELECT */[DISTINCT] Column-name/Expression[Alias]
FROM Table-Name;

Order of execution

1 → FROM

2 → SELECT

Selection: It is used to retrieve data from the table by selecting both rows as well as columns.

Join: It is used to retrieve data from the multiple table simultaneously.

- * Write a query to display all the student names present in the student table.

Select Sname
From Student;

Student					Sname	
Stid	Sname	Branch	Percentage		A	Result table
1	A	MECH	90	→	B	(or)
2	B	CIVIL	85		C	output table
3	C	AERO	87		D	
4	D	ECE	88		E	
5	E	CSE	90		F	
6	F	IT	85			



order execution

output of
From clause

output of
Select clause

order of Execution:-

1 → FROM

2 → SELECT

Notes:-

1. In projection, From clause will execute first.
2. In From clause we can pass table name as an arguments.
3. The job of a From clause is goes to the database & makes the table under execution.
4. After the execution of From clause, Select clause will execute.
5. In a Select clause, we can pass 3 arguments.
 - 1) * Asterisk
 - 2) Column name
 - 3) Expressions
6. A job of a Select clause is goes to an under execution table and search for the column name we have mention.
7. The Select clause is used to fetch the data from the table & display it.
8. Therefore, the Select clause is responsible for Result table (or) output table.

Commands:

- 1) To Set page alignments
 - Set lines 100 pages 100;

- 2) To Clear Screen
 - CLscr (or) Clear Screen

- 3) To describe the table structure
 - desc table_name;
 - desc Student;

• Asterisk * is used to display all the details from the table.

Select *
from table;

• Write a query to display student ID, name, percentage, branch for all the students.

Select * sid, name, per
from student;

• Write a query to display student ID and percentage, branch for all the students.

Select * sid, name, per
from student;

• Write a query to display student ID, student name, branch & percentage for all the students.

Select * sid, name, branch, per
from student;

Asterisk *

It is used to display all the details from table. Along with asterisk we can't use any other column name or expressions.

• Write a query to display all the details present in the student table.

Select *
from student;

• Write a query to display all the details from the employee table.

Select *
from emp;

• Write a query to display all the details from the department table.

Select *
from dept;

• Column names

Employee table	Department table
Emp	Dept
EMPNO	DEPTNO
ENAME	DNAME
JOB	LOC
MGR	
HIREDATE	
SAL	
COMM	
DEPTNO	

1) Write a query to display all the details from the emp.

Select *
from emp;

3) WAGTD Name of all the employees in emp table
 Select name;
 From emp;

4) WAGTD Name of all the employees in emp table
 Select name;
 From emp;

5) WAGTD employee ID & Department number of all the employees in emp table.
 Select empno, deptno
 From emp;

6) WAGTD Name and hiredate of all the employees in emp table.
 Select name, hiredate
 From emp;

7) WAGTD name and designation of all the employees in emp table.
 Select name, job
 From emp;

8) WAGTD name, job & salary given all the employees in emp table.
 Select name, job, sal
 From emp;

9) WAGTD Names present in different table
 Select name
 From dept;

10) WAGTD Names present in different table
 Select name;
 From dept;

* WAGTD Names present in different table
 To remove duplicate values from the result table
 - We can pass distinct as a first argument in the select clause (We cannot use distinct as a second argument in the select clause)
 - To distinct we can use multiple columns. It will remove duplicate or repeating values from the combination of the columns.

* WAGTD Distinct Student Name
 Select Distinct Name
 From Student;

Students

ID	Name	Branch	Per	Roll No.	Grade
1	A	AER	70	A	A
2	B	CIVIL	80	B	B
3	C	MECH	75	C	A
4	D	ECE	75	D	B
5	A	EEE	70	A	A
6	E	CSE	65	E	C
7	B	CIVIL	80	B	B
8	F	BCOM	45	F	F

* WAGTD Distinct Name Branch present in Student table.
 Select Distinct Name, Branch
 From Student;

Sl. No.	Stname	Branch	Per	Stname	Branch	Per	Stname	Branch	Per
1	A	AERO	70	A	AERO	70	A	AERO	70
2	B	CIVIL	80	B	CIVIL	80	B	CIVIL	80
3	C	MECH	75	C	MECH	75	C	MECH	75
4	D	ECE	70	D	ECE	70	D	ECE	70
5	A	EEE	70	A	EEE	70	A	EEE	70
6	E	CSE	65	E	CSE	65	E	CSE	65
7	B	CIVIL	80	B	CIVIL	80	B	CIVIL	80
8	F	BCOM	90	F	BCOM	90	F	BCOM	90

Q.10.10 Different Student name, Branch and Percentage
from student table

Sl. No.	Stname	Branch	Per	Stname	Branch	Per	Stname	Branch	Per
1	A	AERO	70	A	AERO	70	A	AERO	70
2	B	CIVIL	80	B	CIVIL	80	B	CIVIL	80
3	C	MECH	75	C	MECH	75	C	MECH	75
4	D	ECE	70	D	ECE	70	D	ECE	70
5	A	EEE	70	A	EEE	70	A	EEE	70
6	E	CSE	65	E	CSE	65	E	CSE	65
7	B	CIVIL	80	B	CIVIL	80	B	CIVIL	80
8	F	BCOM	90	F	BCOM	90	F	BCOM	90

Q.10.11 Different Branch Per for all student
Select Distinct stname, Branch, Per
from student;

→ Select Stname, distinct Per
from student; — Error

Note: We Cannot use Distinct as a Second argument
in the Select clause

Q.10.12 Different Branch Per for all student
Select Distinct Branch, Per
from student;

Q.10.13 Different Branch Per for all student
Select Distinct Branch, Per
from student;

Q.10.14 Tab Salary and Hiredate for all employee
Select Distinct Tab, Sal, Hiredate
from emp;

Q.10.15 Expression
And Mathematical Input over output as follows
Expression
ex
 $E + 2 * 10 \rightarrow \text{Result}$

Q.10.16 Expression
Expression is always the combination of operators
and operands
Ex: $10 + 20$
operands: may be a character or a number
operators: are the predefined symbols used to
perform specific task

Q.10.17 Annual Salary for all the employees
Select Sal * 12
from emp;

* WAGTS employ name & last name Salary for employees
 \rightarrow select empname, * * *
 from emp;

* WAGTS Salary & salary based on % for all employees
 select Sal, Sal * 12
 from emp;

* WAGTS name, job & salary with deduction of 500 for
 employees
 \rightarrow select name, job, Sal - 500
 from emp;

* WAGTS Salary with hike of 10% for all employees
 select Sal * 1.1, Sal
 from emp;

Note: % with hike for Deduction

$$\frac{\text{Sal} + \text{Sal} \times \frac{K}{100}}{100}$$

$$\frac{\text{Sal} - \text{Sal} \times \frac{K}{100}}{100}$$

* WAGTS Annual Commission & Annual Salary deduction

at 75 % for all emp.

select Comm * 12, Sal * 12 - (Sal * 12 * 75)
 from emp;

$$\frac{\text{Sal} \times 12 + \text{Sal} \times 12 \times \frac{K}{100}}{100}$$

$$\frac{\text{Sal} \times 12 - \text{Sal} \times 12 \times \frac{K}{100}}{100}$$

* WAGTS employees for all employees
 select *

Note: If we select * then all the columns along with Columnname as separator like * * * use tablename.

* [ALTER] is a data manipulation language to alter Columnname or separator in the table.

- \rightarrow Alter can be used to alter table or column name.
- \rightarrow As keyword
- \rightarrow Alter will be a single string
- \rightarrow If it is a multiple string we can use Double quotation (" ") or Underline (_).

select Sal * 12, Sal * 12 - (Sal * 12 * 75)
 from emp;

Eg:

Sal * 12 Annual Salary

Sal * 12 "Annual Salary"

Sal * 12 As "Annual Salary"

Sal * 12 At "Annual Salary"

* WAGTS Annual Salary & Annual Commission with deduction of 100 Rs for All the employees

select Sal * 12 As "Annual sq", Comm * 12 - 100
 At "Annual Comm Deduction"

from emp;

1. SELECT empname, salary, salary * 1.05
FROM emp
WHERE empname < 'M'
ORDER BY salary;

2. SELECT empname, salary, salary * 1.05
FROM emp
WHERE empname < 'M'
ORDER BY salary;

3. SELECT empname;

4. SELECT empname, salary, salary * 1.05
FROM emp
WHERE empname < 'M'
ORDER BY salary;

5. SELECT empname, salary, salary * 1.05
FROM emp
WHERE empname < 'M'
ORDER BY salary;

6. SELECT empname, salary, salary * 1.05
FROM emp
WHERE empname < 'M'
ORDER BY salary;

7. SELECT empname, salary, salary * 1.05
FROM emp
WHERE empname < 'M'
ORDER BY salary;

8. SELECT empname, salary, salary * 1.05
FROM emp
WHERE empname < 'M'
ORDER BY salary;

9. SELECT empname, salary, salary * 1.05
FROM emp
WHERE empname < 'M'
ORDER BY salary;

10. SELECT empname, salary, salary * 1.05
FROM emp
WHERE empname < 'M'
ORDER BY salary;

11. SELECT empname, salary, salary * 1.05
FROM emp
WHERE empname < 'M'
ORDER BY salary;

12. SELECT empname, salary, salary * 1.05
FROM emp
WHERE empname < 'M'
ORDER BY salary;

13. SELECT empname, salary, salary * 1.05
FROM emp
WHERE empname < 'M'
ORDER BY salary;

14. SELECT empname;

15. SELECT empname, salary, salary * 1.05
FROM emp
WHERE empname < 'M'
ORDER BY salary;

16. SELECT empname;

17. SELECT empname, salary, salary * 1.05
FROM emp
WHERE empname < 'M'
ORDER BY salary;

order in which we enter the records
 Hence last records are the from clause
 Hence first records are the to clause
 In SQL 2000 we can use select for loop
 as an example

For example
 To find the maximum of a column Value Date
 To find the minimum of a column Value Date
 To find the maximum of a column Value Date
 To find the minimum of a column Value Date
 To find the maximum of a column Value Date
 To find the minimum of a column Value Date

* WAGTD Name & Design of employees if the
 employees are working in Design K

Select employee, Design
 From emp
 Where Design = 'K'

Output of
 from clause

Empid	Empname	Design
1	JACK	20
2	JAMES	20
3	ALLEN	20
4	MILLER	20
5	WARD	20
6	SMITH	20
7	KING	20

under execution

After execution
 Design = K

1	JACK	20
2	JAMES	20
3	ALLEN	20
4	MILLER	20
5	WARD	20
6	SMITH	20
7	KING	20

output of Where

Empid	Empname	Design
1	JACK	20
4	MILLER	10
7	KING	10

output of
 select clause

Empname	Design
JAMES	20
ALLEN	10
KING	10

* WAGTD Name & salary of the employees if employee
 are working in any design that less
 than 25000

Select empname, sal
 From emp
 Where sal < 25000

* WAGTD Details of an employee Where Working in
 a manager

Select *
 From emp
 Where Job = 'MANAGER'

* WAGTD Name, Job & HireDate of employees if employee
 are hired after 21st Dec 82

Select empname, Job, HireDate
 From emp
 Where HireDate > '21-DEC-82'

* WAGTD Details of employee along with annual sal
 if annual sal is less than 25000

Select emp.*, Sal * 12 as "Annual Salary"
 From emp
 Where Sal * 12 < 25000

Note:- We cannot use Alias Name in Where clause
 Because Where clause executes before select clause

1. Select empid, empname, salary, deptname
 from emp, dept
 where emp.deptid = dept.deptid

2. Select empid, empname, salary, deptname
 from emp, dept
 where emp.deptid = dept.deptid
 and emp.salary > 1000

3. Select empid, empname, salary, deptname
 from emp, dept
 where emp.deptid = dept.deptid
 and emp.hiredate > '2000-01-01'

4. Select empid, empname, salary, deptname
 from emp, dept
 where emp.deptid = dept.deptid
 and emp.salary > 1000

5. Select empid, empname, salary, deptname
 from emp, dept
 where emp.deptid = dept.deptid
 and emp.salary > 1000

Assignment

1. Select empid, empname, salary, deptname
 from emp, dept
 where emp.deptid = dept.deptid

2. Select empid, empname, salary, deptname
 from emp, dept
 where emp.deptid = dept.deptid
 and emp.salary > 1000

3. Select empid, empname, salary, deptname
 from emp, dept
 where emp.deptid = dept.deptid
 and emp.hiredate > '2000-01-01'

4. Select empid, empname, salary, deptname
 from emp, dept
 where emp.deptid = dept.deptid
 and emp.salary > 1000

5. Select empid, empname, salary, deptname
 from emp, dept
 where emp.deptid = dept.deptid
 and emp.hiredate > '2000-01-01'

6. Select empid, empname, salary, deptname
 from emp, dept
 where emp.deptid = dept.deptid
 and emp.salary > 1000

12. WAGTS NAME and SALARY of Emps. who have been in the annual salary more than 10000.
 Select empname, salary as Annual-Sal
 from emp
 where salary > 10000.

13. WAGTS EMPID of the employees who are working in DEPT 3.
 Select empid
 from emp
 where deptno = 3.

14. WAGTS EMPNAME and EMPID of Emps who are working in 1991.
 Select empname, empid
 from emp
 where hiredate < '01-JAN-91';

15. WAGTS Details of emp whose salary is more than 10000.
 Select *
 from emp
 where salary > 10000.

16. WAGTS NAME and SALARY of Emps who are working in DEPT 3.
 Select empname, salary
 from emp
 where deptno = 3.

17. WAGTS DETAILS of Emps who are working in DEPT 3.
 Select *
 from emp
 where deptno = 3;

18. WAGTS EMPID of employees who are working in DEPT 3.
 Select empid
 from emp
 where deptno = 3.

19. WAGTS Details of employees who are working in DEPT 3.
 Select *
 from emp
 where deptno = 3.

20. WAGTS Details of Emps who are working in DEPT 3.
 Select *
 from emp
 where deptno = 3.

- 1. Arithmetic operators: (+, -, *, /)
- 2. Concatenation operators: ||
- 3. Comparison operators: (=, <, >, <=, >=, <>, <=, >=)
- 4. Relational operators: <, >, <=, >=, <>, <=, >=
- 5. Logical operators: (AND, OR, NOT)
- 6. Special operators: (||, ||)

1. NOT IN
2. BETWEEN
3. NOT BETWEEN
4. IS
5. IS NOT
6. LIKE
7. NOT LIKE
8. ALL
9. ANY
10. EXISTS
11. NOT EXISTS

The given string:
The symbol for concatenation operator is

||

From EMP:

MR || ENAME

MR || ENAME

MR || ENAME

SMITH || ENAME ||

Select 'MR' || ENAME || ' || ENAME || ' || ENAME ||

From EMP;

Select 'MR' || ENAME || ' || ENAME || ' || ENAME ||

From EMP;

Where ENAME = 'SMITH';

It is used to write multiple conditions in a where clause.
AND, OR, NOT.

It is a Binary Multiplication operator.

- AND operator return true if all the conditions get satisfied.

get Satisfied

It is a binary operator, it is also known as logical operator.
NOT operator is similar to Negation.
It is used to negate the condition.

Define have

Not Define

WADT Details of

Select *

From emp

Where Job = 'MANAGER' AND DEPT =

WADT

Select

Select

From emp;

Where Job = 'Salesman' AND Sal > 1000 AND Dept = 3;

```

SELECT empname, job,
       FROM emp
WHERE job = 'analyst' OR job = 'analyst';

```

```

SELECT empname, job,
       FROM emp
WHERE job = 'analyst' OR job = 'analyst';

```

```

SELECT empname, job,
       FROM emp
WHERE job = 'analyst' OR job = 'analyst';

```

```

SELECT empname, job,
       FROM emp
WHERE job = 'analyst' OR job = 'analyst';

```

```

SELECT empname, job,
       FROM emp
WHERE job = 'analyst' OR job = 'analyst';

```

```

SELECT empname, job,
       FROM emp
WHERE job = 'analyst' OR job = 'analyst';

```

```

SELECT empname, job,
       FROM emp
WHERE job = 'analyst' OR job = 'analyst';

```

```

SELECT empname, job,
       FROM emp
WHERE job = 'analyst' OR job = 'analyst';

```

```

SELECT empname, job,
       FROM emp
WHERE job = 'analyst' OR job = 'analyst';

```

```

SELECT empname, job,
       FROM emp
WHERE job = 'analyst' OR job = 'analyst';

```

```

SELECT empname, job,
       FROM emp
WHERE job = 'analyst' OR job = 'analyst';

```

SELECT * FROM emp
 WHERE empno = 7369

SELECT * FROM emp
 WHERE empno = 7369

SELECT * FROM emp
 WHERE empno = 7369

SELECT * FROM emp
 WHERE empno = 7369

SELECT * FROM emp
 WHERE empno = 7369

SELECT * FROM emp
 WHERE empno = 7369

SELECT * FROM emp
 WHERE empno = 7369

SELECT * FROM emp
 WHERE empno = 7369

SELECT * FROM emp
 WHERE empno = 7369

SELECT * FROM emp
 WHERE empno = 7369

SELECT * FROM emp
 WHERE empno = 7369

SELECT * FROM emp
 WHERE empno = 7369

1. The first step is to identify the problem or question that needs to be answered. This involves understanding the context and the specific requirements of the task.

$\sum_{i=1}^n x_i = 1$
 For $i=1, 2, \dots, n$
 $x_i \geq 0$

[illegible]

10/1/1919

1. General
 2. Specific
 3. Particular
 4. Detail
 5. Example
 6. Illustration
 7. Comparison
 8. Contrast
 9. Analogy
 10. Metaphor
 11. Simile
 12. Personification
 13. Hyperbole
 14. Irony
 15. Sarcasm
 16. Allegory
 17. Symbolism
 18. Imagery
 19. Onomatopoeia
 20. Personification
 21. Hyperbole
 22. Irony
 23. Sarcasm
 24. Allegory
 25. Symbolism
 26. Imagery
 27. Onomatopoeia
 28. Personification
 29. Hyperbole
 30. Irony
 31. Sarcasm
 32. Allegory
 33. Symbolism
 34. Imagery
 35. Onomatopoeia
 36. Personification
 37. Hyperbole
 38. Irony
 39. Sarcasm
 40. Allegory
 41. Symbolism
 42. Imagery
 43. Onomatopoeia
 44. Personification
 45. Hyperbole
 46. Irony
 47. Sarcasm
 48. Allegory
 49. Symbolism
 50. Imagery
 51. Onomatopoeia
 52. Personification
 53. Hyperbole
 54. Irony
 55. Sarcasm
 56. Allegory
 57. Symbolism
 58. Imagery
 59. Onomatopoeia
 60. Personification
 61. Hyperbole
 62. Irony
 63. Sarcasm
 64. Allegory
 65. Symbolism
 66. Imagery
 67. Onomatopoeia
 68. Personification
 69. Hyperbole
 70. Irony
 71. Sarcasm
 72. Allegory
 73. Symbolism
 74. Imagery
 75. Onomatopoeia
 76. Personification
 77. Hyperbole
 78. Irony
 79. Sarcasm
 80. Allegory
 81. Symbolism
 82. Imagery
 83. Onomatopoeia
 84. Personification
 85. Hyperbole
 86. Irony
 87. Sarcasm
 88. Allegory
 89. Symbolism
 90. Imagery
 91. Onomatopoeia
 92. Personification
 93. Hyperbole
 94. Irony
 95. Sarcasm
 96. Allegory
 97. Symbolism
 98. Imagery
 99. Onomatopoeia
 100. Personification
 101. Hyperbole
 102. Irony
 103. Sarcasm
 104. Allegory
 105. Symbolism
 106. Imagery
 107. Onomatopoeia
 108. Personification
 109. Hyperbole
 110. Irony
 111. Sarcasm
 112. Allegory
 113. Symbolism
 114. Imagery
 115. Onomatopoeia
 116. Personification
 117. Hyperbole
 118. Irony
 119. Sarcasm
 120. Allegory
 121. Symbolism
 122. Imagery
 123. Onomatopoeia
 124. Personification
 125. Hyperbole
 126. Irony
 127. Sarcasm
 128. Allegory
 129. Symbolism
 130. Imagery
 131. Onomatopoeia
 132. Personification
 133. Hyperbole
 134. Irony
 135. Sarcasm
 136. Allegory
 137. Symbolism
 138. Imagery
 139. Onomatopoeia
 140. Personification
 141. Hyperbole
 142. Irony
 143. Sarcasm
 144. Allegory
 145. Symbolism
 146. Imagery
 147. Onomatopoeia
 148. Personification
 149. Hyperbole
 150. Irony
 151. Sarcasm
 152. Allegory
 153. Symbolism
 154. Imagery
 155. Onomatopoeia
 156. Personification
 157. Hyperbole
 158. Irony
 159. Sarcasm
 160. Allegory
 161. Symbolism
 162. Imagery
 163. Onomatopoeia
 164. Personification
 165. Hyperbole
 166. Irony
 167. Sarcasm
 168. Allegory
 169. Symbolism
 170. Imagery
 171. Onomatopoeia
 172. Personification
 173. Hyperbole
 174. Irony
 175. Sarcasm
 176. Allegory
 177. Symbolism
 178. Imagery
 179. Onomatopoeia
 180. Personification
 181. Hyperbole
 182. Irony
 183. Sarcasm
 184. Allegory
 185. Symbolism
 186. Imagery
 187. Onomatopoeia
 188. Personification
 189. Hyperbole
 190. Irony
 191. Sarcasm
 192. Allegory
 193. Symbolism
 194. Imagery
 195. Onomatopoeia
 196. Personification
 197. Hyperbole
 198. Irony
 199. Sarcasm
 200. Allegory
 201. Symbolism
 202. Imagery
 203. Onomatopoeia
 204. Personification
 205. Hyperbole
 206. Irony
 207. Sarcasm
 208. Allegory
 209. Symbolism
 210. Imagery
 211. Onomatopoeia
 212. Personification
 213. Hyperbole
 214. Irony
 215. Sarcasm
 216. Allegory
 217. Symbolism
 218. Imagery
 219. Onomatopoeia
 220. Personification
 221. Hyperbole
 222. Irony
 223. Sarcasm
 224. Allegory
 225. Symbolism
 226. Imagery
 227. Onomatopoeia
 228. Personification
 229. Hyperbole
 230. Irony
 231. Sarcasm
 232. Allegory
 233. Symbolism
 234. Imagery
 235. Onomatopoeia
 236. Personification
 237. Hyperbole
 238. Irony
 239. Sarcasm

1st camp
2nd camp
3rd camp
4th camp
5th camp
6th camp
7th camp
8th camp
9th camp
10th camp
11th camp
12th camp
13th camp
14th camp
15th camp
16th camp
17th camp
18th camp
19th camp
20th camp
21st camp
22nd camp
23rd camp
24th camp
25th camp
26th camp
27th camp
28th camp
29th camp
30th camp
31st camp
32nd camp
33rd camp
34th camp
35th camp
36th camp
37th camp
38th camp
39th camp
40th camp
41st camp
42nd camp
43rd camp
44th camp
45th camp
46th camp
47th camp
48th camp
49th camp
50th camp
51st camp
52nd camp
53rd camp
54th camp
55th camp
56th camp
57th camp
58th camp
59th camp
60th camp
61st camp
62nd camp
63rd camp
64th camp
65th camp
66th camp
67th camp
68th camp
69th camp
70th camp
71st camp
72nd camp
73rd camp
74th camp
75th camp
76th camp
77th camp
78th camp
79th camp
80th camp
81st camp
82nd camp
83rd camp
84th camp
85th camp
86th camp
87th camp
88th camp
89th camp
90th camp
91st camp
92nd camp
93rd camp
94th camp
95th camp
96th camp
97th camp
98th camp
99th camp
100th camp

1. The first step is to identify the problem or question that needs to be answered. This involves understanding the context and the specific requirements of the task.

1. 1944
 2. 1945
 3. 1946
 4. 1947
 5. 1948
 6. 1949
 7. 1950
 8. 1951
 9. 1952
 10. 1953
 11. 1954
 12. 1955
 13. 1956
 14. 1957
 15. 1958
 16. 1959
 17. 1960
 18. 1961
 19. 1962
 20. 1963
 21. 1964
 22. 1965
 23. 1966
 24. 1967
 25. 1968
 26. 1969
 27. 1970
 28. 1971
 29. 1972
 30. 1973
 31. 1974
 32. 1975
 33. 1976
 34. 1977
 35. 1978
 36. 1979
 37. 1980
 38. 1981
 39. 1982
 40. 1983
 41. 1984
 42. 1985
 43. 1986
 44. 1987
 45. 1988
 46. 1989
 47. 1990
 48. 1991
 49. 1992
 50. 1993
 51. 1994
 52. 1995
 53. 1996
 54. 1997
 55. 1998
 56. 1999
 57. 2000
 58. 2001
 59. 2002
 60. 2003
 61. 2004
 62. 2005
 63. 2006
 64. 2007
 65. 2008
 66. 2009
 67. 2010
 68. 2011
 69. 2012
 70. 2013
 71. 2014
 72. 2015
 73. 2016
 74. 2017
 75. 2018
 76. 2019
 77. 2020
 78. 2021
 79. 2022
 80. 2023
 81. 2024
 82. 2025
 83. 2026
 84. 2027
 85. 2028
 86. 2029
 87. 2030
 88. 2031
 89. 2032
 90. 2033
 91. 2034
 92. 2035
 93. 2036
 94. 2037
 95. 2038
 96. 2039
 97. 2040
 98. 2041
 99. 2042
 100. 2043
 101. 2044
 102. 2045
 103. 2046
 104. 2047
 105. 2048
 106. 2049
 107. 2050
 108. 2051
 109. 2052
 110. 2053
 111. 2054
 112. 2055
 113. 2056
 114. 2057
 115. 2058
 116. 2059
 117. 2060
 118. 2061
 119. 2062
 120. 2063
 121. 2064
 122. 2065
 123. 2066
 124. 2067
 125. 2068
 126. 2069
 127. 2070
 128. 2071
 129. 2072
 130. 2073
 131. 2074
 132. 2075
 133. 2076
 134. 2077
 135. 2078
 136. 2079
 137. 2080
 138. 2081
 139. 2082
 140. 2083
 141. 2084
 142. 2085
 143. 2086
 144. 2087
 145. 2088
 146. 2089
 147. 2090
 148. 2091
 149. 2092
 150. 2093
 151. 2094
 152. 2095
 153. 2096
 154. 2097
 155. 2098
 156. 2099
 157. 2100
 158. 2101
 159. 2102
 160. 2103
 161. 2104
 162. 2105
 163. 2106
 164. 2107
 165. 2108
 166. 2109
 167. 2110
 168. 2111
 169. 2112
 170. 2113
 171. 2114
 172. 2115
 173. 2116
 174. 2117
 175. 2118
 176. 2119
 177. 2120
 178. 2121
 179. 2122
 180. 2123
 181. 2124
 182. 2125
 183. 2126
 184. 2127
 185. 2128
 186. 2129
 187. 2130
 188. 2131
 189. 2132
 190. 2133
 191. 2134
 192. 2135
 193. 2136
 194. 2137
 195. 2138
 196. 2139
 197. 2140
 198. 2141
 199. 2142
 200. 2143
 201. 2144
 202. 2145
 203. 2146
 204. 2147
 205. 2148
 206. 2149
 207. 2150
 208. 2151
 209. 2152
 210. 2153
 211. 2154
 212. 2155
 213. 2156
 214. 2157
 215. 2158
 216. 2159
 217. 2160
 218. 2161
 219. 2162
 220. 2163
 221. 2164

Johnston, N.H. 1890-1891

$$\sum_{j=0}^{\infty} \frac{1}{(n+j)^2} = \frac{1}{n} + O\left(\frac{1}{n^2}\right) \quad (n \rightarrow \infty).$$

in order to square them
in order. It knows all the other operators
and can multiply them. It says to the
other operators to do it.

The first part of the paper is devoted to the study of the
 asymptotic behavior of the solutions of the system (1) as
 $t \rightarrow \infty$. It is shown that the solutions of the system (1)
 tend to zero as $t \rightarrow \infty$ if and only if the matrix
 A is stable.

Name: _____
 Date: _____
 Subject: _____



1948-1949

3. 6. 1 - 12. 6. 1946

1. 45. 7. 1951

1947-1948
1949-1950

$\frac{1}{2} \left(\frac{1}{2} \right) = \frac{1}{4}$

$$\begin{aligned} \frac{1}{2} \cdot 10 &= 5 \\ 10000 &= 10000 \\ 10000 &= 10000 \end{aligned}$$

$\frac{1}{2} \left(\frac{1}{2} \right) = \frac{1}{4}$

4. 5. 1. 8

4. $\alpha \in \mathbb{C} \setminus \mathbb{R}$
 5. $\alpha \in \mathbb{R}$ and $\alpha \neq 0$ and $\alpha \neq 1$ and $\alpha \neq -1$ and $\alpha \neq \frac{1}{2}$ and $\alpha \neq -\frac{1}{2}$

[illegible]

15. Staphylococcus aureus - commonest cause of skin infection
features like abscess, furuncle, carbuncle

Streptococcus is a chain of 10

Between operators when the way is clear,
the correct interchange the beam is not required.

2. The number of people who are not in the club is 100 - 40 = 60.

$$T_{\text{eff}} = T \left(1 - \frac{\alpha}{\beta} \right) \quad \text{for } \alpha < \beta$$
 $\hat{c} = c_{\text{eff}}(t) \quad \hat{\omega} = \omega(t)$

1. $2\text{Al} + 3\text{H}_2\text{SO}_4 \rightarrow \text{Al}_2(\text{SO}_4)_3 + 3\text{H}_2$

emp, dept
 emp, dept
 emp, dept
 emp, dept
 emp, dept
 emp, dept

select * from emp
 where dept = 10
 select * from emp
 where dept = 10

select *
 from emp
 where hiredate between '01-JAN-81' and '31-DEC-81';
 select * from emp
 where hiredate between '01-JAN-81' and '31-DEC-81';
 select * from emp
 where hiredate between '01-JAN-81' and '31-DEC-81';

select * from emp
 where hiredate between '01-JAN-81' and '31-DEC-81';
 select * from emp
 where hiredate between '01-JAN-81' and '31-DEC-81';

Column name / expression NOT BETWEEN value and value

select *
 from emp
 where sal not between 1250 and 1500;

select *
 from emp
 where sal not between 1250 and 1500;

select *
 from emp
 where hiredate not between '01-JAN-81' and '31-DEC-81';

... to compare with only the employees
... to compare with only the employees

Select empname, comm
From emp
Where comm is null;

Select
From emp
Where comm is null;

Ex: To select the employees with not null values

Select
Empname, empcomm IS NOT NULL;

Select
From emp
Where Comm is not null and Comm less;

Select
From emp
Where comm is not null and Comm is null;

... to compare with only the employees
... to compare with only the employees

Select empname, empcomm
From emp
Where comm is null;

Select
From emp
Where comm is null;

Ex: To select the employees with not null values

Select empname
From emp
Where empname like 'A%';

Select
From emp
Where empname like 'A%';

Select empname
From emp
Where empname like 'A%' and 'B%';

1. Select *
From emp
Where name like 'A%'

2. Select *
From emp
Where name like 'A%'

3. Select *
From emp
Where name like 'A%'

4. Select *
From emp
Where name like 'A%'

5. Select *
From emp
Where name like 'A%'

6. Select *
From emp
Where name like 'A%'

7. Select *
From emp
Where name like 'A%'

8. Select *
From emp
Where name like 'A%'

9. Select *
From emp
Where name like 'A%'

10. Select *
From emp
Where name like 'A%'

11. Select *
From emp
Where name like 'A%'

12. Select *
From emp
Where name like 'A%'

13. Select *
From emp
Where name like 'A%'

14. Select *
From emp
Where name like 'A%'

1. Select empname, hiredate
from emp
where empname like 'A%';

2. Select empname, hiredate
from emp
where hiredate like '%Feb%';

3. Select empname, sal
from emp
where sal like '5000';

4. Select empname, sal
from emp
where sal like '5000' and hiredate like '%Feb%';

5. Select empname, sal
from emp
where sal like '5000' and empname like 'A%';

6. Select empname
from emp
where empname like 'A%' and hiredate like '%Feb%';

7. Select empname, hiredate
from emp
where hiredate between '1980-01-01' and '1980-01-31';

8. Select empname, sal
from emp
where hiredate between '1980-01-01' and '1980-01-31';

9. Select empname, sal
from emp
where empname is not null;

10. Select empname, sal
from emp
where empname is not null and hiredate like '%Feb%';

11. Select empname, sal
from emp
where empname is not null and hiredate like '%Feb%' and sal like '5000';

12. Select empname, sal
from emp
where empname is not null and hiredate like '%Feb%' and sal like '5000' and empname like 'A%';

13. Select empname, hiredate
from emp
where hiredate between '1980-01-01' and '1980-01-31' and empname like 'A%';

1. List the names of employees working as Analysts
during 1989 in Dept 10

Select *
From emp
Where Job = 'Analyst' and Sal > 1000

2. List the names of the employees who have been hired
during 1989

Select empname
From emp
Where Hiredate Like '%1989%'

3. List the details of the employees working at Salem and
Dallas, earn salary commission

Select *
From emp
Where Job = 'Salesman' and (Location = 'Salem' or Location = 'Dallas')

4. List the Name and Salary Given to the emp hired
during 1989 in Dept 10 in 1989

Select empname, Sal
From emp
Where Hiredate between '01-Jan-1989' and '31-Dec-1989'
Deptno = 10

5. List the details of emp having 1's Present in the name
and works as Manager

Select *
From emp
Where empname Like '%1%' and Job = 'Manager'

6. List the names of employees who have been hired
during 1989

Select empname
From emp
Where Hiredate between '01-Jan-1989' and '31-Dec-1989'

7. List the names of the emp of name length 10
and salary

Select empname
From emp
Where empname Like '_____' and Sal > 1000

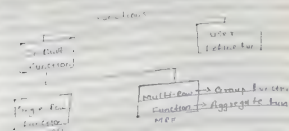
8. List All the emp Names who get 1's in emp name
Name has 'a' At the 3rd position

Select empname
From emp
Where Name Like '%_a_%'

9. List the details of the emp working at Salem
and hired After 1984 and has a Name length 10

Select *
From emp
Where (Job = 'Manager' and Hiredate > '31-Dec-1984'
AND empname Like '%10%')

Function - the set of instructions to perform specific task.



101	1000	100
102	1000	100
103	1000	100
104	1000	100

101	1000	100
102	1000	100
103	1000	100
104	1000	100

- If it receives data by row
- If we pass one input or multiple input function, it processes and generates one output.
- If we pass multiple inputs or single row function, it generates multiple outputs.

Emp	
NAME	100
ALLEN	1000
MILLER	3000
PERC	1000
MARKER	4000
PERC	2000

Length: To count the number of characters.
 Select Empname, Length(Empname) from Emp;

Emp	
NAME	100
ALLEN	1000
MILLER	3000
PERC	1000
MARKER	4000
PERC	2000

- Multi-Row Function / Aggregate Function
- Multi-Row Function Aggregates all the data and then it processes and generates one output.
- If we pass one input or multiple inputs, it generates one output.
- If we pass multiple inputs or single row function, it generates one output.

List of Multi-Row Function

- Max()
- Min()
- Sum()
- Avg()
- Count()

Example: max sal present in emp table
 select Max(sal) from emp;

Output
 Max(sal)
 4700



- Rules of rules for function
- Multi param we can use Single args (input)
- The arguments may be of name of expression
- Along with them we cannot use any Col name as expression
- merge() ignores the null values
- The values are distributed in where clause
- Count is the only function that takes aliases
- Asubstr() as a argument

select emp_id, emp_name, emp_sal
from emp;

select emp_id, emp_name
from emp;

select emp_id, emp_name, emp_sal
from emp;

select emp_id, emp_name, emp_sal
from emp;

select emp_id, emp_name, emp_sal
from emp;

select emp_id, emp_name, emp_sal
from emp;

Col name / expression like/was like 'Pattern to match' (Case Sensitive)

1. Escape character must be defined
2. Escape char must be used before the special character (% or -) which has to be treated as a normal character.
3. The recommended character for escape is: [\ / \ \ \]

Index	Column	Value
100	emp_id	100
101	emp_name	John
102	emp_sal	10000
103	emp_sal	10000
104	emp_sal	10000
105	emp_sal	10000
106	emp_sal	10000
107	emp_sal	10000
108	emp_sal	10000
109	emp_sal	10000
110	emp_sal	10000
111	emp_sal	10000
112	emp_sal	10000
113	emp_sal	10000
114	emp_sal	10000
115	emp_sal	10000
116	emp_sal	10000
117	emp_sal	10000
118	emp_sal	10000
119	emp_sal	10000
120	emp_sal	10000
121	emp_sal	10000
122	emp_sal	10000
123	emp_sal	10000
124	emp_sal	10000
125	emp_sal	10000
126	emp_sal	10000
127	emp_sal	10000
128	emp_sal	10000
129	emp_sal	10000
130	emp_sal	10000
131	emp_sal	10000
132	emp_sal	10000
133	emp_sal	10000
134	emp_sal	10000
135	emp_sal	10000
136	emp_sal	10000
137	emp_sal	10000
138	emp_sal	10000
139	emp_sal	10000
140	emp_sal	10000
141	emp_sal	10000
142	emp_sal	10000
143	emp_sal	10000
144	emp_sal	10000
145	emp_sal	10000
146	emp_sal	10000
147	emp_sal	10000
148	emp_sal	10000
149	emp_sal	10000
150	emp_sal	10000
151	emp_sal	10000
152	emp_sal	10000
153	emp_sal	10000
154	emp_sal	10000
155	emp_sal	10000
156	emp_sal	10000
157	emp_sal	10000
158	emp_sal	10000
159	emp_sal	10000
160	emp_sal	10000
161	emp_sal	10000
162	emp_sal	10000
163	emp_sal	10000
164	emp_sal	10000
165	emp_sal	10000
166	emp_sal	10000
167	emp_sal	10000
168	emp_sal	10000
169	emp_sal	10000
170	emp_sal	10000
171	emp_sal	10000
172	emp_sal	10000
173	emp_sal	10000
174	emp_sal	10000
175	emp_sal	10000
176	emp_sal	10000
177	emp_sal	10000
178	emp_sal	10000
179	emp_sal	10000
180	emp_sal	10000
181	emp_sal	10000
182	emp_sal	10000
183	emp_sal	10000
184	emp_sal	10000
185	emp_sal	10000
186	emp_sal	10000
187	emp_sal	10000
188	emp_sal	10000
189	emp_sal	10000
190	emp_sal	10000
191	emp_sal	10000
192	emp_sal	10000
193	emp_sal	10000
194	emp_sal	10000
195	emp_sal	10000
196	emp_sal	10000
197	emp_sal	10000
198	emp_sal	10000
199	emp_sal	10000

steel tank
 from 20 to
 1000 Tons of steel

Inland Trans. like 1970-71

1. The first part of the paper is devoted to a discussion of the
 2. second part of the paper is devoted to a discussion of the
 3. third part of the paper is devoted to a discussion of the
 4. fourth part of the paper is devoted to a discussion of the
 5. fifth part of the paper is devoted to a discussion of the

selected several
from each
where Gold Leaf and Dettol in:

1942-1943
 1944-1945
 1946-1947
 1948-1949
 1950-1951
 1952-1953
 1954-1955
 1956-1957
 1958-1959
 1960-1961
 1962-1963
 1964-1965
 1966-1967
 1968-1969
 1970-1971
 1972-1973
 1974-1975
 1976-1977
 1978-1979
 1980-1981
 1982-1983
 1984-1985
 1986-1987
 1988-1989
 1990-1991
 1992-1993
 1994-1995
 1996-1997
 1998-1999
 2000-2001
 2002-2003
 2004-2005
 2006-2007
 2008-2009
 2010-2011
 2012-2013
 2014-2015
 2016-2017
 2018-2019
 2020-2021
 2022-2023
 2024-2025
 2026-2027
 2028-2029
 2030-2031
 2032-2033
 2034-2035
 2036-2037
 2038-2039
 2040-2041
 2042-2043
 2044-2045
 2046-2047
 2048-2049
 2050-2051
 2052-2053
 2054-2055
 2056-2057
 2058-2059
 2060-2061
 2062-2063
 2064-2065
 2066-2067
 2068-2069
 2070-2071
 2072-2073
 2074-2075
 2076-2077
 2078-2079
 2080-2081
 2082-2083
 2084-2085
 2086-2087
 2088-2089
 2090-2091
 2092-2093
 2094-2095
 2096-2097
 2098-2099
 2100-2101
 2102-2103
 2104-2105
 2106-2107
 2108-2109
 2110-2111
 2112-2113
 2114-2115
 2116-2117
 2118-2119
 2120-2121
 2122-2123
 2124-2125
 2126-2127
 2128-2129
 2130-2131
 2132-2133
 2134-2135
 2136-2137
 2138-2139
 2140-2141
 2142-2143
 2144-2145
 2146-2147
 2148-2149
 2150-2151
 2152-2153
 2154-2155
 2156-2157
 2158-2159
 2160-2161
 2162-2163
 2164-2165
 2166-2167
 2168-2169
 2170-2171
 2172-2173
 2174-2175
 2176-2177
 2178-2179
 2180-2181
 2182-2183
 2184-2185
 2186-2187
 2188-2189
 2190-2191
 2192-2193
 2194-2195
 2196-2197
 2198-2199
 2200-2201
 2202-2203
 2204-2205
 2206-2207
 2208-2209
 2210-2211
 2212-2213
 2214-2215
 2216-2217
 2218-2219
 2220-2221
 2222-2223
 2224-2225
 2226-2227
 2228-2229
 2230-2231
 2232-2233
 2234-2235
 2236-2237
 2238-2239
 2240-2241
 2242-2243
 2244-2245
 2246-2247
 2248-2249
 2250-2251
 2252-2253
 2254-2255
 2256-2257
 2258-2259
 2260-2261
 2262-2263
 2264-2265
 2266-2267
 2268-2269
 2270-2271
 2272-2273
 2274-2275
 2276-2277
 2278-2279
 2280-2281
 2282-2283
 2284-2285
 2286-2287
 2288-2289
 2290-2291
 2292-2293
 2294-2295
 2296-2297
 2298-2299
 2300-2301
 2302-2303
 2304-2305
 2306-2307
 2308-2309
 2310-2311
 2312-2313
 2314-2315
 2316-2317
 2318-2319
 2320-2321
 2322-2323
 2324-2325
 2326-2327
 2328-2329
 2330-2331
 2332-2333
 2334-2335
 2336-2337
 2338-2339
 2340-2341
 2342-2343
 2344-2345
 2346-2347
 2348-2349
 2350-2351
 2352-2353
 2354-2355
 2356-2357
 2358-2359
 2360-2361
 2362-2363
 2364-2365
 2366-2367
 2368-2369
 2370-2371
 2372-2373
 2374-2375
 2376-2377
 2378-2379
 2380-2381
 2382-2383
 2384-2385
 2386-2387
 2388-2389
 2390-2391
 2392-2393
 2394-2395
 2396-2397
 2398-2399
 2400-2401
 2402-2403
 2404-2405
 2406-2407
 2408-2409
 2410-2411
 2412-2413
 2414-2415
 2416-2417
 2418-2419
 2420-2421
 2422-2423
 2424-2425
 2426-2427
 2428-2429
 2430-2431
 2432-2433
 2434-2435
 2436-2437
 2438-2439
 2440-2441
 2442-2443
 2444-2445
 2446-2447
 2448-2449
 2450-2451
 2452-2453
 2454-2455
 2456-2457
 2458-2459
 2460-2461
 2462-2463
 2464-2465
 2466-2467
 2468-2469
 2470-2471
 2472-2473
 2474-2475
 2476-2477
 2478-2479
 2480-2481
 2482-2483
 2484-2485
 2486-2487
 2488-2489
 2490-2491
 2492-2493
 2494-2495
 2496-2497
 2498-2499
 2500-2501
 2502-2503
 2504-2505
 2506-2507
 2508-2509
 2510-2511
 2512-2513
 2514-2515
 2516-2517
 2518-2519
 2520-2521
 2522-2523
 2524-2525
 252

def. 1. Avg. of normal to top of comp.
 2. sec. Avg. (Sec.)
 From comp.

$$\begin{aligned} \frac{1}{x} &= x^{-1} \quad \frac{d}{dx} x^{-1} = -1x^{-2} = -\frac{1}{x^2} \\ \frac{1}{x^2} &= x^{-2} \quad \frac{d}{dx} x^{-2} = -2x^{-3} = -\frac{2}{x^3} \\ \frac{1}{x^3} &= x^{-3} \quad \frac{d}{dx} x^{-3} = -3x^{-4} = -\frac{3}{x^4} \end{aligned}$$

2000-2001
 2001-2002
 2002-2003
 2003-2004
 2004-2005
 2005-2006
 2006-2007
 2007-2008
 2008-2009
 2009-2010
 2010-2011
 2011-2012
 2012-2013
 2013-2014
 2014-2015
 2015-2016
 2016-2017
 2017-2018
 2018-2019
 2019-2020
 2020-2021
 2021-2022
 2022-2023
 2023-2024
 2024-2025
 2025-2026
 2026-2027
 2027-2028
 2028-2029
 2029-2030
 2030-2031
 2031-2032
 2032-2033
 2033-2034
 2034-2035
 2035-2036
 2036-2037
 2037-2038
 2038-2039
 2039-2040
 2040-2041
 2041-2042
 2042-2043
 2043-2044
 2044-2045
 2045-2046
 2046-2047
 2047-2048
 2048-2049
 2049-2050
 2050-2051
 2051-2052
 2052-2053
 2053-2054
 2054-2055
 2055-2056
 2056-2057
 2057-2058
 2058-2059
 2059-2060
 2060-2061
 2061-2062
 2062-2063
 2063-2064
 2064-2065
 2065-2066
 2066-2067
 2067-2068
 2068-2069
 2069-2070
 2070-2071
 2071-2072
 2072-2073
 2073-2074
 2074-2075
 2075-2076
 2076-2077
 2077-2078
 2078-2079
 2079-2080
 2080-2081
 2081-2082
 2082-2083
 2083-2084
 2084-2085
 2085-2086
 2086-2087
 2087-2088
 2088-2089
 2089-2090
 2090-2091
 2091-2092
 2092-2093
 2093-2094
 2094-2095
 2095-2096
 2096-2097
 2097-2098
 2098-2099
 2099-2100
 2100-2101
 2101-2102
 2102-2103
 2103-2104
 2104-2105
 2105-2106
 2106-2107
 2107-2108
 2108-2109
 2109-2110
 2110-2111
 2111-2112
 2112-2113
 2113-2114
 2114-2115
 2115-2116
 2116-2117
 2117-2118
 2118-2119
 2119-2120
 2120-2121
 2121-2122
 2122-2123
 2123-2124
 2124-2125
 2125-2126
 2126-2127
 2127-2128
 2128-2129
 2129-2130
 2130-2131
 2131-2132
 2132-2133
 2133-2134
 2134-2135
 2135-2136
 2136-2137
 2137-2138
 2138-2139
 2139-2140
 2140-2141
 2141-2142
 2142-2143
 2143-2144
 2144-2145
 2145-2146
 2146-2147
 2147-2148
 2148-2149
 2149-2150
 2150-2151
 2151-2152
 2152-2153
 2153-2154
 2154-2155
 2155-2156
 2156-2157
 2157-2158
 2158-2159
 2159-2160
 2160-2161
 2161-2162
 2162-2163
 2163-2164
 2164-2165
 2165-2166
 2166-2167
 2167-2168
 2168-2169
 2169-2170
 2170-2171
 2171-2172
 2172-2173
 2173-2174
 2174-2175
 2175-2176
 2176-2177
 2177-2178
 2178-2179
 2179-2180
 2180-2181
 2181-2182
 2182-2183
 2183-2184
 2184-2185
 2185-2186
 2186-2187
 2187-2188
 2188-2189
 2189-2190
 2190-2191
 2191-2192
 2192-2193
 2193-2194
 2194-2195
 2195-2196
 2196-2197
 2197-2198
 2198-2199
 2199-2200
 2200-2201
 2201-2202
 2202-2203
 2203-2204
 2204-2205
 2205-2206
 2206-2207
 2207-2208
 2208-2209
 2209-2210
 2210-2211
 2211-2212
 2212-2213
 2213-2214
 2214-2215
 2215-2216
 2216-2217
 2217-2218
 2218-2219
 2219-2220
 2220-2221
 2221-2222
 2222-2223
 2223-2224
 2224-2225
 2225-2226
 2226-2227
 2227-2228
 2228-2229
 2229-2230
 2230-2231
 2231-2232
 2232-2233
 2233-2234
 2234-2235
 2235-2236
 2236-2237
 2237-2238
 2238-2239
 2239-2240
 2240-2241
 2241-2242
 2242-2243
 2243-2244
 2244-2245
 2245-2246
 2246-2247
 2247-2248
 2248-2249
 2249-2250
 2250-2251
 2251-2252
 2252-2253
 2253-2254
 2254-2255
 2255-2256
 2256-2257
 2257-2258
 2258-2259
 2259-2260
 2260-2261
 2261-2262
 2262-2263
 2263-2264
 2264-2265
 2265-2266
 2266-2267
 2267-2268
 2268-2269
 2269-2270
 2270-2271
 2271-2272
 2272-2273
 2273-2274
 2274-2275
 2275-2276
 2276-2277
 2277-2278
 2278-2279
 2279-2280
 2280-2281
 2281-2282
 2282-2283
 2283-2284
 2284-2285
 2285-2286
 2286-2287
 2287-2288
 2288-2289
 2289-2290
 2290-2291
 2291-2292
 229

Subject: Geographical
 Exam: and
 Where: hydrological

$\Delta f = \frac{1}{2} \Delta f_{\text{max}}$
 Select Count &
 Loss amp
 where max 74.5

Subject Counted
from cup
Where Comm is Martin and Centre St.

Select Anglo-Saxon, ^{Scandinavian} Germanic, Slavonic
 from camp
 influence John "President"

SELECT emp1
FROM emp1
WHERE emp1.emp1 = 'John'

SELECT emp1
FROM emp1
WHERE emp1.emp1 = 'John'

SELECT emp1
FROM emp1
WHERE emp1.emp1 = 'John'

SELECT emp1
FROM emp1
WHERE emp1.emp1 = 'John'

SELECT emp1
FROM emp1
WHERE emp1.emp1 = 'John'

SELECT emp1
FROM emp1
WHERE emp1.emp1 = 'John'

SELECT emp1
FROM emp1
WHERE emp1.emp1 = 'John'

SELECT emp1
FROM emp1
WHERE emp1.emp1 = 'John'

SELECT emp1
FROM emp1
WHERE emp1.emp1 = 'John'

SELECT emp1
FROM emp1
WHERE emp1.emp1 = 'John'

SELECT emp1
FROM emp1
WHERE emp1.emp1 = 'John'

SELECT * FROM EMP WHERE SAL > 1000

SELECT * FROM EMP WHERE SAL > 1000
 SELECT * FROM EMP WHERE SAL > 1000
 SELECT * FROM EMP WHERE SAL > 1000

SELECT * FROM EMP

- FROM
- WHERE (if used)
- ORDER BY
- GROUP BY

- GROUP BY clause is used to group the records
- It creates two by two
- group by clause we can use as column/Expression as argument
- After the execution of group by clause the data is given
- Therefore, group by clause creates the group by
- it will execute group by group
- Without where clause we can use group by clause

SELECT * FROM EMP WHERE SAL > 1000

Select * From Emp
 Group by Deptno

EMPNO	ENAME	JOB	MANAGER	STARTDATE	ENDDATE	DEPTNO
1	ALLEN	SALES	10	1981-02-20	1981-05-19	20
2	JAMES	SALES	10	1981-01-23	1981-04-19	20
3	MILLER	SALES	10	1981-02-24	1981-04-19	20
4	BLAKE	MANAGER	10	1981-03-24	1981-04-19	20
5	CLARK	SALES	10	1981-06-09	1981-04-19	20
6	FORD	ANALYST	10	1981-01-03	1981-04-19	20
7	SCOTT	ANALYST	10	1981-07-06	1981-04-19	20

GROUP BY

DEPTNO	AVG(SAL)
20	1500
30	1600
40	1700
50	1800
60	1900
70	2000
80	2100
90	2200

GROUP BY

3. GROUP BY clause is used to the employees in each dept.
select empno, last_name
from emp
group by dept;

4. GROUP BY clause is used to the employees in each dept.
select empno, last_name, department_id
from emp
where dept = 'Marketing Sales Dept'
group by dept;

5. GROUP BY clause is used to the employees in each dept.
select empno, last_name
from emp
where dept = 'Marketing Sales Dept'
group by dept;

6. GROUP BY clause is used to the employees in each dept.
select Avg(sal) as AvgSal
from emp
where dept = 'Marketing Sales Dept'
group by dept;

7. GROUP BY clause is used to the employees in each dept.
select empno, last_name, department_id
from emp
where dept = 'Marketing Sales Dept'
group by dept;

8. GROUP BY clause is used to the employees in each dept.
select empno, last_name, department_id
from emp
where dept = 'Marketing Sales Dept'
group by dept;

9. GROUP BY clause is used to the employees in each dept.
select empno, last_name, department_id
from emp
where dept = 'Marketing Sales Dept'
group by dept;

10. GROUP BY clause is used to the employees in each dept.
select empno, last_name, department_id
from emp
where dept = 'Marketing Sales Dept'
group by dept;

11. GROUP BY clause is used to the employees in each dept.
select empno, last_name, department_id
from emp
where dept = 'Marketing Sales Dept'
group by dept;

12. GROUP BY clause is used to the employees in each dept.
select empno, last_name, department_id
from emp
where dept = 'Marketing Sales Dept'
group by dept;

select * from emp where sal > 1000
 select * from emp where sal > 1000
 select * from emp where sal > 1000
 select * from emp where sal > 1000
 select * from emp where sal > 1000

select * from emp where sal > 1000
 select * from emp where sal > 1000
 select * from emp where sal > 1000
 select * from emp where sal > 1000

select * from emp where sal > 1000
 select * from emp where sal > 1000
 select * from emp where sal > 1000
 select * from emp where sal > 1000

select * from emp where sal > 1000
 select * from emp where sal > 1000
 select * from emp where sal > 1000
 select * from emp where sal > 1000

select * from emp where sal > 1000

select * from emp where sal > 1000
 select * from emp where sal > 1000
 select * from emp where sal > 1000
 select * from emp where sal > 1000

select * from emp where sal > 1000
 select * from emp where sal > 1000
 select * from emp where sal > 1000
 select * from emp where sal > 1000

Emp	NAME	LASTNAME	DATE	TIME	DATE	TIME
1	ALLEN	1966	10	10	10	10
2	BAKER	1950	10	10	10	10
3	JAMES	1960	10	10	10	10
4	MILLER	1960	10	10	10	10
5	FORD	1960	10	10	10	10
6	ADAMS	1960	10	10	10	10
7	TURNER	1960	10	10	10	10
8	KING	1960	10	10	10	10
9	CLARK	1960	10	10	10	10
10	MARTIN	1960	10	10	10	10
11	BLAKE	1960	10	10	10	10
12	SCOTT	1960	10	10	10	10
13	FORD	1960	10	10	10	10
14	TURNER	1960	10	10	10	10
15	ADAMS	1960	10	10	10	10
16	SCOTT	1960	10	10	10	10
17	FORD	1960	10	10	10	10
18	TURNER	1960	10	10	10	10
19	ADAMS	1960	10	10	10	10
20	SCOTT	1960	10	10	10	10

1. 1. 1.

These cups are breaking at very low pressure

$\frac{d}{dt} \left(\frac{1}{r^2} \right) = -\frac{2}{r^3} \frac{dr}{dt}$

Select (undated) for records of
 from. CMB
 Given by Ben
 Moving Records 1961

```

1. Initial Set with all values of  $\alpha$ 
    Select  $\alpha_{min}$  and
    from emp
    Group by Set
    Having  $\alpha_{min}(\alpha) > 1$ ;

```

select empid
 from emp
 where empid > 100
 order by empid

select empid, last_name
 from emp
 where empid > 100
 order by empid
 Having count(*) > 2

select empid, last_name
 from emp
 where empid > 100
 order by empid
 Having count(*) > 2

select empid, last_name
 from emp
 where empid > 100
 order by empid
 Having count(*) > 2

select empid, last_name
 from emp
 where empid > 100
 order by empid

select empid, last_name
 from emp
 where empid > 100
 order by empid
 Having count(*) > 2

select empid, last_name
 from emp
 where empid > 100
 order by empid
 Having count(*) > 2

select empid, last_name
 from emp
 where empid > 100
 order by empid
 Having count(*) > 2

1. **SELECT** clause is used to select the records from the table.
 2. **FROM** clause is used to select the records from the table.
 3. **WHERE** clause is used to select the records from the table.
 4. **ORDER BY** clause is used to sort the records in ascending or descending order.
 5. **GROUP BY** clause is used to group the records.
 6. **HAVING** clause is used to filter the records.
 7. **JOIN** clause is used to join the records from two or more tables.
 8. **UNION** clause is used to combine the records from two or more tables.
 9. **EXCEPT** clause is used to subtract the records from one table from another table.
 10. **INTERSECT** clause is used to select the common records from two or more tables.
 11. **AS** clause is used to give an alias to the table or column.
 12. **NULL** is a special value that represents the absence of any data.
 13. **NOT NULL** is a constraint that requires a column to have a value.
 14. **PRIMARY KEY** is a constraint that uniquely identifies each record in a table.
 15. **FOREIGN KEY** is a constraint that establishes a relationship between two tables.
 16. **INDEX** is a data structure that improves the speed of retrieval operations on a database.
 17. **VIEW** is a virtual table that is created from the result of a query.
 18. **TRIGGER** is a stored procedure that is automatically executed when a certain event occurs in the database.
 19. **SEQUENCE** is a database object that generates a series of unique numbers.
 20. **SYNONYM** is a database object that is an alias for another database object.

1. **SELECT** clause is used to select the records from the table.
 2. **FROM** clause is used to select the records from the table.
 3. **WHERE** clause is used to select the records from the table.
 4. **ORDER BY** clause is used to sort the records in ascending or descending order.
 5. **GROUP BY** clause is used to group the records.
 6. **HAVING** clause is used to filter the records.
 7. **JOIN** clause is used to join the records from two or more tables.
 8. **UNION** clause is used to combine the records from two or more tables.
 9. **EXCEPT** clause is used to subtract the records from one table from another table.
 10. **INTERSECT** clause is used to select the common records from two or more tables.
 11. **AS** clause is used to give an alias to the table or column.
 12. **NULL** is a special value that represents the absence of any data.
 13. **NOT NULL** is a constraint that requires a column to have a value.
 14. **PRIMARY KEY** is a constraint that uniquely identifies each record in a table.
 15. **FOREIGN KEY** is a constraint that establishes a relationship between two tables.
 16. **INDEX** is a data structure that improves the speed of retrieval operations on a database.
 17. **VIEW** is a virtual table that is created from the result of a query.
 18. **TRIGGER** is a stored procedure that is automatically executed when a certain event occurs in the database.
 19. **SEQUENCE** is a database object that generates a series of unique numbers.
 20. **SYNONYM** is a database object that is an alias for another database object.

* WAGDA Max Sal given to Emps and Deptno at least 2 emp are working in each department Arrange the records in dep wise

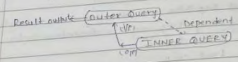
```

Select Max(sal), Deptno
from emp
Group By Deptno
Having Count(*) >= 2
Order By Deptno;

```

* Subquery: A query return inside another query in known as Subquery.

Interlocking Flow:-



- Let us consider into query outer query & inner query
- In Subquery inner query will execute first and generates an output.
- The output of an inner query is sending as an input to the outer query
- With the help of this input outer query will execute and generates an result and an output table.
- Therefore outer query is dependent on inner query.

* When and why we use Subquery?

Case 1
When there is unknown condition present in a query we use subquery.

```

Emp:
Empno Sal Deptno
ADAMS 1500 10
JONES 3800 30
BLAKE 4000 10
FORD 4000 10
KING 1200 30

```

Select Empno

* WAGDA Names of Emps who are earning Sal greater than 1500

```

Select Empno
from Emp
where Sal > 1500;

```

Direct Condition
or
Known Condition

* WAGDA Names of Emps who are earning Sal greater than Emp.

```

Select Empno
from Emp
where Sal > (Select Sal
from Emp
where Empno = 'JONES');

```

Indirect Condition
or
Unknown Condition

Sal	F
1200	F
1500	F
3800	T
4000	T
1200	F

JONES (3800, 10)

* WAQTD Name & Hiredate of an emp if Emps are hired before the Miller

```

Select empname, hiredate
from emp
where hiredate < (Select hiredate
from emp
where empname = 'Miller');

```

* WAQTD Names of employees who are working in the same designation as Adams

```

Select empname
from emp
where Job = (Select Job
from emp
where empname = 'Adams');

```

* WAQTD Details of Emps who are working in the same dept as Jones

```

Select *
from emp
where deptno = (Select deptno
from emp
where empname = 'JONES');

```

* WAQTD Details of Emps who are working as a manager and hired after the Smith

```

Select *
from emp
where Job = 'Manager' AND hiredate >
(Select hiredate
from emp
where empname = 'Smith');

```

* WAQTD Name & Sal of Emps if Emps are Earning Sal less than the Blake and emp name has Char A

```

Select empname, Sal
from emp
where empname like 'A%' AND Sal < (Select Sal
from emp
where empname = 'Blake');

```

* WAQTD Details of emp who are Earning Sal greater than Jones and less than King

```

Select *
from emp
where Sal > (Select Sal from emp
where empname = 'Jones') AND Sal < (Select Sal
from emp
where empname = 'King');

```

* WAQTD Details of Emp who are hired after Adams before a Adams & Emp hired year of 81

```

Select *
from emp
where hiredate > (Select hiredate
from emp
where empname = 'ADAMS') AND hiredate < (Select hiredate
from emp
where empname = 'ADAMS');

```

* WAQTD Details of Emp along with Annual Sal if Emps are working in same dept as Turner and same designation as Martin and Earning Sal greater than Smith

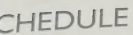
```

Select emp#, Sal * 12
from emp
where deptno = (Select deptno
from emp
where empname = 'Smith');

```

Where empno = 'Turner' AND Job = (Select empno
from emp
where Empno = 'MARTIN')

Sel = (Select Sal
from emp
where Empno = 'SMITH');



WEDNESDAY

THURSDAY

FRIDAY

my self

Name: _____

Amco Rez

Class:

School / College:

Email:

Index

Sign

Remark

Subject:

Date _____

Page
No.